86805: Software Architectures for Robotics
Localization system for a wheeled humanoid robot

Date: Wed 23:55, 01.02.16

Prof. F. Mastrogiovanni, PhD C. Recchiuto

Rabbia Asghar, BEng, Ernest Skrzypczyk, BSc

Contents

Rollo - Humanoid robot

Odometry

Odometry is currently the most widely used technique for determining the position of a mobile robot. It mainly involves use of various encoders, for example on wheels, as sensors to estimate the robot's position relative to a starting or previous location. Usually, it is used for real-time positioning in the between the periodic absolute position measurements, for example GPS (Global Positioning System) provides absolute position feedback, however it updates at $0.1 \div 1s$ interval, during which odometry could be used for localization. One of the major downsides of odometry is its sensitivity to errors. There are various error sources discussed in section Odometry errors on page 4 in detail. One significant source of error influencing the accuracy of odometry that is worth mentioning however, is the integration of velocity measurements over time to give position estimates.

First the odometry based motion model for the robot will be derived.

The model is derived based on the following important assumptions:

1. The robot is a rigid body
2. The model represents a differential drive robot
3. There is no slip in the wheels
4. Both wheels are turning in the forward direction

A differential drive robot runs straight when the linear speed of both the left and right wheel is same. If the speed of one wheel is greater than the other, the robot runs in an arc. This derivation can be divided in three distinct cases of robot motion:

The basic premise for the odometry model of the Rollo humanoid robot is presented in figures (**??**) and (**??**).

1. Clockwise direction
2. Counterclockwise direction
3. Straight line

Clockwise direction

First, the case is considered when the speed of left wheel is greater than the right, and the robot will run in clockwise direction. Both right and left wheel will rotate around the same center of a circle.

P_i represents the initial position of the robot, defined by the center of the line joining two wheels, while l represents axle length, ergo distance between two wheels. S_L and S_R represent the distance travelled by left and right wheel respectively. Θ represents the angle of travel for both the wheels and r is the radius of travel from the center of robot. With encoder feedback from left and right wheel, S_L and S_R can simply be computed using equations (1) and (2)

$$S_L = \frac{n_L}{60} 2\pi r_L \tag{1}$$

$$S_R = \frac{n_R}{60} 2\pi r_R \tag{2}$$

where n_L and n_R are the revolutions per minute ([rpm]) of left and right wheel, and r_L and r_R are the radii of the 2 wheels.

S_L and S_R can be related to r and Θ using formulas (3) and (4)

$$S_L = (r + \frac{l}{2})\Theta \tag{3}$$

$$S_R = (r - \frac{l}{2})\Theta \tag{4}$$

The equations (3) and (4) on page ?? can be solved simultaneously to compute r and l.

$$r = \frac{l}{2} \cdot \frac{S_L + S_R}{S_L - S_R} \tag{5}$$

$$\Theta = \frac{S_R}{r - \frac{l}{2}} \tag{6}$$

Final position of the robot and its orientation can then be derived using basic trigonometry and geometry relations as displayed in equation (7):

$$P_f = \begin{bmatrix} P_{ix} + r(1 - \cos(\Theta)), & P_{iy} + r\sin(\Theta) \end{bmatrix} \tag{7}$$

$$\text{final orienation} = \text{initial orientation} - \Theta$$

Counterclockwise direction

Similar derivation of the robot can be derived for counterclockwise rotation, when the rpm of right wheel is higher than that of the left wheel. The relations for final position and rotation are as follows for this case:

$$P_f = \begin{bmatrix} P_{ix} - r(1 - \cos(\Theta)), & P_{iy} + r\sin(\Theta) \end{bmatrix} \tag{8}$$

$$\text{final orienation} = \text{initial orientation} + \Theta$$

Straight line motion

$$P_f = \begin{bmatrix} P_{ix} + S_L(\cos(\Theta)), & P_{iy} + S_L\sin(\Theta) \end{bmatrix} \tag{9}$$

$$\text{final orienation} = \text{initial orientation}$$

Adaption of Odometry Model for Rollo

Currently, the encoders feedback is unavailable in the robot and true odometry model can not be implemented. However, an attempt has been made to implement its modified version. The distance covered by right and left wheels is instead estimated from the control command.

Straight line motion

Because of the mechanical difference in the two legs, it is almost impossible for the two wheels to run at same speed . Moreover, the two motors/wheels are powered using 2 different LIPO packs and are run in open loop. The difference in voltage level further results in different speed of the motor at the same pwm. This is a major source for deviation in behavior of robot's adjusted odometry model.

Rollo was run in a straight line at different commands and its location feedback from the motion capture system was logged. For a certain command, average speed of left and right wheel was determined from the logs.

Odometry errors

This method is sensitive to errors Rapid and accurate data collection, equipment calibration, and processing are required in most cases for odometry to be used effectively straight forward to implement

Odometry Error Sources:

1. Limited resolution during integration (time increments, measurement resolution).
2. Unequal wheel diameter (deterministic)
3. Variation in the contact point of the wheel (deterministic)
4. Unequal floor contact and variable friction can lead to slipping (non deterministic)

The errors can be divided into systematic and random. Three main sources of systematic errors in odometry:

- Distance
- Rotation
- Skew

Last two more significant with time.

System and Measurements Model

System Model

In order to implement Kalman Filter, the previously described model must be first represented in state space representation.

We can define the location of the robot at instant k using state variables. This will include position in x and y coordinates and orientation.

$$x_k = \begin{bmatrix} x_{[x],k} \\ x_{[y],k} \\ x_{[\phi],k} \end{bmatrix} \tag{10}$$

The control input provided to robot is the speed of right and left wheel. This defines the distance covered by both the wheels in unit time. The relative displacement of the robot at instant k can be notated by d_k. Using equations 8,7 and 9, relative displacement can be expressed in terms of r and θ. Thus, control input u_k can be expressed as a function of relative displacement.

$$u_k = j(d_k) \tag{11}$$

$$u_k = \begin{bmatrix} u_{[r],k} \\ u_{[\phi],k} \end{bmatrix}$$

Given x_{k-1} and u_{k-1}, the next location of the robot, x_k can be computed.

$$x_k = f(x_{k-1}, u_{k-1}) = \begin{bmatrix} f_x(x_{k-1}, u_{k-1}) \\ f_y(x_{k-1}, u_{k-1}) \\ f_\phi(x_{k-1}, u_{k-1}) \end{bmatrix} \tag{12}$$

In the above derivation of the system model it was assumed that there are no noise sources. In the next section, we model the noise in the system.

System Model with noise

We assume that the noise in the odometry can be modeled by a random noise vector q_k such that the noise is Gaussian distribution with zero mean,$\hat{q_k}$ and covariance matrix,U_k .

$$q_k \sim N(\hat{q_k}, U_k) \tag{13}$$

where

$$\hat{q_k} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

and

$$U_k = \mathrm{E}(q_k - \hat{q_k})(q_k - \hat{q_k})^T$$

$$U_k = \begin{bmatrix} \sigma^2_{q[r],k} & \sigma_{q[\phi],k}\sigma_{q[r],k} \\ \sigma_{q[\phi],k}\sigma_{q[r],k} & \sigma^2_{q[\phi],k} \end{bmatrix}$$

With the assumption that the noise sources are independent, the off-diagonal elements of the covariance matrix, U_k are equal to zero.The computation of variances $\sigma^2_{q[r],k}$ and $\sigma^2_{q[\phi],k}$ for the model is discussed in section.

The control input, or relative displacement can be expressed now as shown below.

$$u_k = j(d_k) + q_k \tag{14}$$

$$u_k = \begin{bmatrix} u_{[r],k} \\ u_{[\phi],k} \end{bmatrix} + \begin{bmatrix} q_{[r],k} \\ q_{[\phi],k} \end{bmatrix}$$

This makes u_k a random vector. Assuming, that $u_{[r],k}$ and $u_{[\phi],k}$ are deterministic, uncertainty in$u_k$equals the uncertainty in the noise term q_k.

The system noise can similarly be modeled by a random noise vector w_k such that the noise is Gaussian distribution with zero mean , $\hat{w_k}$ and covariance matrix, Q_k

$$w_k \sim N(\hat{w_k}, Q_k) \tag{15}$$

where

$$\hat{w_k} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

and

$$Q_k = \mathrm{E}(w_k - \hat{w_k})(w_k - \hat{w_k})^T$$

$$Q_k = \begin{bmatrix} \sigma^2_{w[x],k} & \sigma_{w[y],k}\sigma_{w[x],k} & \sigma_{w[\phi],k}\sigma_{w[x],k} \\ \sigma_{w[x],k}\sigma_{w[y],k} & \sigma^2_{w[y],k} & \sigma_{w[\phi],k}\sigma_{w[y],k} \\ \sigma_{w[x],k}\sigma_{w[\phi],k} & \sigma_{w[\phi],k}\sigma_{w[x],k} & \sigma^2_{w[\phi],k} \end{bmatrix}$$

With the assumption that the noise sources are independent, the off-diagonal elements of the covariance matrix, Q_k are equal to zero.The computation of variances $\sigma^2_{w[x],k}$, $\sigma^2_{w[y],k}$ and $\sigma^2_{w[\phi],k}$ for the model is discussed in section.

The system can be expressed now as shown below.

$$x_k = f(x_{k-1}, u_{k-1}) + w_k \tag{16}$$

$$x_k = \begin{bmatrix} f_x(x_{k-1}, u_{k-1}) \\ f_y(x_{k-1}, u_{k-1}) \\ f_\phi(x_{k-1}, u_{k-1}) \end{bmatrix} + \begin{bmatrix} w_{[x],k-1}) \\ w_{[y],k-1}) \\ w_{[\phi],k-1}) \end{bmatrix} \tag{17}$$

w_k consists of noise sources that are not directly related to u_k. Now, x_k is a random vector and with every time step the system noise increases the variance. Thus, the variance of the location grows with every time step.

We have proposed the System Model so that we can implement Kalman Filter on it. Now, we need to prepare the Measurement Model for it.

Measurement Model with noise

Starting with the assumption that there is no noise in the measurement, z_k, it is simply a vector containing for each state variable a variable that takes on the value of the corresponding state variable. The measurement vector, z_k is

$$z_k = \begin{bmatrix} z_{[x],k} \\ z_{[y],k} \\ z_{[\phi],k} \end{bmatrix} \tag{18}$$

In our system, we use the motion capture system in the lab to localize the robot. The motion capture system acts as absolute sensor and provides us directly the position of the robot in x and y coordinates and its orientation. Thus, z_k in this case is simply expressed as 19

$$z_k = \begin{bmatrix} x_{[x],k} \\ x_{[y],k} \\ x_{[\phi],k} \end{bmatrix}$$

Now, we can add noise to the measurement model. We assume that the noise in the odometry can be modeled by a random noise vector v_k such that the noise is Gaussian distribution with zero mean,$\hat{v_k}$ and covariance matrix,R_k .

$$v_k \sim N(\hat{v_k}, R_k) \tag{19}$$

where

$$\hat{v_k} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

and

$$R_k = \mathrm{E}(v_k - \hat{v_k})(v_k - \hat{v_k})^T$$

$$R_k = \begin{bmatrix} \sigma^2_{v[x],k} & \sigma_{v[y],k}\sigma_{v[x],k} & \sigma_{v[\phi],k}\sigma_{v[x],k} \\ \sigma_{v[x],k}\sigma_{v[y],k} & \sigma^2_{v[y],k} & \sigma_{v[\phi],k}\sigma_{v[y],k} \\ \sigma_{v[x],k}\sigma_{v[\phi],k} & \sigma_{v[\phi],k}\sigma_{v[x],k} & \sigma^2_{v[\phi],k} \end{bmatrix}$$

With the assumption that the noise sources are independent, the off-diagonal elements of the covariance matrix, R_k are equal to zero.

The measurement model can be expressed now as shown below.

$$z_k = x_k + v_k \tag{20}$$

$$z_k = \begin{bmatrix} x_{[x],k} \\ x_{[y],k} \\ x_{[\phi],k} \end{bmatrix} + \begin{bmatrix} v_{[x],k-1}) \\ v_{[y],k-1}) \\ v_{[\phi],k-1}) \end{bmatrix}$$

Since the measurement noise v_k is a Gaussian vector, this makes z_k a random vector.

Conclusions